IFW AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

SWAINE et al

Atty. Ref.: 550-205; Confirmation No. 3984

Appl. No. 09/773,387

TC/A.U. 2192

Filed: February 1, 2001

Examiner: J. Derek Rutten

For: TRACING OUT-OF-ORDER DATA

* * * * * * * * * * * *

December 27, 2005

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF (37 CRF 41.37)

In response to the Notification of Non-Compliant Appeal Brief dated November 28, 2005, Appellants submit herewith a new Brief that responds to the objections raised by the Examiner. Specifically, the summary section of the Brief now includes specific page and line number references for the subject matter defined in the independent claims. In addition, the corresponding structure for the "means" elements recited in independent claim 14 from the specification has been identified. As explained in the Appeal Brief, the means elements are not limited to the corresponding structure identified but also include all equivalents of that corresponding identified structure. Although the Examiner suggests that the summary be limited to a just one-for-one correspondence of claim elements with reference numerals and corresponding specification text, the rules do not prohibit Appellants from explaining the context

of the subject matter of the claimed invention. Indeed, it is important for the Board Members to understand that context, including the problems recognized by the inventors and the solutions they invented to overcome those problems.

Entry of the Appeal Brief is respectfully requested.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: _____
John R. Lastova
Reg. No. 33,149

JRL:sd
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

SWAINE et al

Serial No. 09/773,387

Filed: February 1, 2001

For: TRACING OUT-OF-ORDER DATA

Atty. Ref.: 550-205

Group: 2192

Examiner: James Rutten

Before the Board of Patent Appeals and Interferences

BRIEF FOR APPELLANT

On Appeal From Final Rejection
From Group Art Unit 2192

John R. Lastova
NIXON & VANDERHYE P.C.
11th Floor, 901 North Glebe Road
Arlington, Virginia 22203-1808
(703) 816-4025
Attorney for Appellants
Swaine et al
ARM Limited

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of

SWAINE et al — Atty. Ref.: 550-205

Serial No. 09/773,387 — Group: 2192

Filed: February 1, 2001 — Examiner: James Rutten

For: TRACING OUT-OF-ORDER DATA

* *

December 27, 2005

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

I. REAL PARTY IN INTEREST

The real party in interest is the assignee, ARM Limited, a United Kingdom corporation.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals related to this subject application. There are no interferences related to this subject application.

III. STATUS OF CLAIMS

Claims 1-24 are pending. Claims 1-4, 7, 12-17, and 20 stand rejected under 35 U.S.C. §103 as being unpatentable over the description of the prior art, (referred as to "DPA"), in view of USP 6,681,321 to Dale et al. Claims 5 and 18 stand rejected under 35 U.S.C. §103 as being unpatentable over DPA in view of Dale et al. and further in view of USP 5,555,392 to Chaput et al. Claims 6 and 19 stand rejected under 35 U.S.C. §103 as being unpatentable over DPA in view of Dale et al., in view of USP 5,555,392 to Chaput, and further in view of USP 6,009,270 to Mann. Claims 8-10 and 21-23 stand rejected under 35 U.S.C. §103 as being unpatentable over DPA in view of Dale et al. and further in view of EPA 0 465 765 A2 to Westcott. Claims 11 and 24 stand rejected under 35 U.S.C. §103 as being unpatentable over DPA in view of Dale et al., further in view of Westcott, and further in view of USP 5,978,742 to Pickerd.

IV. STATUS OF AMENDMENTS

No amendment has been filed after final.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The claims are directed to data processing systems providing tracing mechanisms to enable data accesses to be traced. See, e.g., page 4, lines 5-22. Tracing the activity of a data processing system by generating a trace stream representing the step-by-step activity within the system is a highly useful tool in system development. For more deeply embedded processor cores, it is more difficult to track the state of the core via

externally accessible pins. Accordingly, tracing functionality is being placed on-chip for capturing and analyzing trace data. See, e.g., page 1, lines 10-21.

Independent claim 1 recites a processing circuit, a memory, a tracing circuit, and a data access instruction. Figure 1 illustrates a non-limiting example of a data processing system 2 providing a on-chip tracing mechanism. An integrated circuit 4 includes a microprocessor core 6, a cache memory 8, an on-chip trace module controller 10, and an on-chip trace buffer 12. See, e.g., page 10, lines 7-9. The integrated circuit 4 is connected to an external memory 14 which is accessed when a cache miss occurs within the cache memory 8. Id. at lines 9-10. A general purpose computer 16 is coupled to the on-chip trace module controller 10 and the on-chip trace buffer 12 and serves to recover and analyze a stream of tracing data from these elements using software executing upon the general purpose computer 16. *Id.* at lines 10-13.

Traditional tracing encounters problems when different instructions may be executed in parallel, in a pipeline fashion, or in a manner where the completion of an instruction is not always required before the next instruction can be commenced. An example is a response to a "load miss." Page 1, lines 23-page 2, line 8. A load miss can occur when a data processing system seeks to load a data value (which may be data for processing or an instruction word) from a memory location. Claim 1 recites that a data access instruction may result in a data miss. If that data value is present within a local cache memory, then the load instruction may be completed rapidly, possibly in a single clock cycle. However, should a load miss occur where the data value load cannot be satisfied from the cache and requires a slower non-cache access, such as to a main memory, then the data value will not be returned for possibly many processing cycles.

Rather than halt data processing, it is known to provide systems, such as the ARM1020T processor, in which other program instructions can continue to execute while waiting for the data from the previous load miss, provided that those later instructions do not require or depend on the data value that has not yet been retrieved. The inventors recognized a serious problem in this context: how to provide meaningful tracing when an instruction stream being executed does not easily correlate with the data values being returned from memory accesses as observed on the memory buses. Page 3, lines 10-25.

Figure 2 (shown below) illustrates example responses to data access instructions that may occur within the system of Figure 1. The first code sequence illustrates a load instruction 18 which loads data into a register from the memory location specified by an address given in another register R_m. In this case, the data access instruction results in a hit within the cache memory 8 and the corresponding data value from the address specified by the register R_m is returned on the same cycle to the processor core 6. Page 11, lines 1-6.

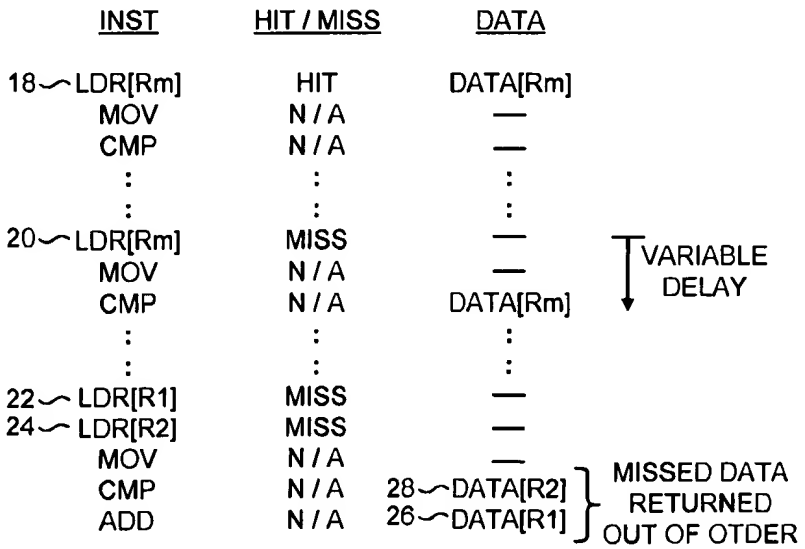


FIG. 2

For the second time the same instruction sequence is issued, the instruction 20 results in a miss within the cache memory 8. The integrated circuit 4 continues to execute the instructions following the load instruction 20 that gave rise to the miss, providing those subsequent instructions do not require the missed data. At a variable delay time later, the data corresponding to the access miss is returned on the data bus of the system. The late data will be routed to the correct register. Page 11, lines 8-14. The third code sequence in Figure 2 has two load instructions 22, 24 that both result in data misses and have corresponding late data returned. The late data is returned on the data bus out of order from the sequence of the data access instructions 22, 24 that gave rise to it. Accordingly, the first missing data access 22 corresponds to the second late data item 26, and the second missing data access 24 corresponds to the first late data item 28. Page 11, line 24-page 12, line 4.

To solve these problems with data access misses and late returned data values, a data place holder is inserted within the stream of trace data identifying the point at which the data value would have been returned if the miss had not occurred and then later inserting a late data value resulting from the miss. Claim 1 describes generating a data place holder and inserting a late data value. The data place holder and the late data value enable subsequent data stream analysis to correctly identify which data values correspond to which access instructions and so obtain a proper understanding of the data processing system behavior.

The correlation between the data place holders and the late data may be achieved in various different ways. One non-limiting example way illustrated in Figure 3 (reproduced below) is to associate a tag value with each data place holder (claim 3 recites

a tag value) and then associate a corresponding tag value with each late data value (this mechanism is also able to cope with late data items being returned in a different order to their respective place holders).

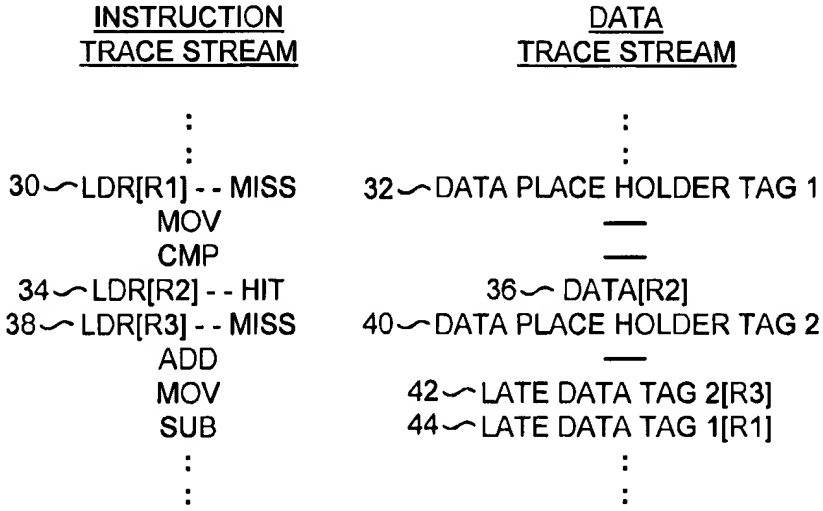


FIG. 3

Figure 3 illustrates the behavior of the on-chip trace module controller 10 in response to data access instructions that give rise to data misses. The on-chip trace module 10 writes its trace data at high speed into the on-chip trace buffer 12. This enables the tracing to occur in real time keeping pace with the full speed execution of instructions by the processor core 6. At a later time the general purpose computer 16 may read the contents of the trace buffer 12 to analyze the trace data. Page 12, lines 6-11. Thus, upon analysis of the stream of trace data, the respective tag values can be matched to determine which late data value corresponds to which data place holder.

In the specific example shown in Figure 3, a first load instruction 30 results in a data miss within the cache memory 8. A data place holder 32 is inserted into the data trace stream at the corresponding point for the instruction 30 within the instruction trace

stream that gave rise to the miss. This data place holder 32 has an associated tag value (Tag1) that identifies that data place holder. At a later time, a further data accessing instruction 34 results in a hit with the data value 36 being returned in the same processing cycle and placed within the data trace stream at a point directly matching the instruction 34. The next instruction 38 results in a further data miss and the insertion of a further data place holder 40 with a different identifying tag value (Tag2). Page 12, lines 13-22.

At a variable time later, two late data values 42, 44 are returned on the data bus and inserted into the data trace stream. Each of these late data values has a tag value associated with it that enables it to be matched to a corresponding data place holder earlier within the data trace stream. In this particular example, the late data values are returned out of order with the data misses that gave rise to them. Accordingly, the first late data value 42 returned has a tag value of Tag2 and corresponds to the data place holder 40 and the instruction 38. The second late data value 44 has a tag value of Tag1 and corresponds to the data place holder 32 and the instruction 30. Page 12, line 23-page 13, line 6.

Another example (see Figure 4) is one in which when each data place holder is inserted in the stream of traced data an indication is also given as to how many pending late data values are outstanding at that time such that when the stream of traced data is later analyzed the appropriate late data value can be matched to the appropriate data place holder without requiring a complete stream of trace data. A further example (see Figure 5) is one in which the stream of traced data includes periodic synchronizing data indicating how many outstanding late data values are awaited at that time. Claim 6 relates to periodic synchronizing data. Thus, when the stream of traced data is picked up

partway through, once a synchronizing data portion had been read, then subsequent data place holders and data values can be matched together.

Claim 14 includes claim elements recited using means-plus-function format. As required by the rules, the following example structure disclosed in the application corresponding to those means elements is identified. Of course, the "means" elements are not limited to the corresponding structure disclosed in the specification but also encompass in any and all equivalents to the disclosed structure. Structure from two different embodiments illustrated in Figures 1 and 9 is identified.

The means for processing includes, for example, core 6 in Figure 1, or in Figure 9, the register bank 108, the multiplier 112, the shifter 114, the adder 116, and the instruction decoder 118. The means for storing data and corresponds to the cache memory 8 in Figure 1 or the cache memory 110 in Figure 9. The means for generating a stream of trace data includes the on-chip trace module 10 in Figure 1 or the trace controller on 20 in Figure 9. The means for storing also may include the external memory 14 and/or external storage 18 shown in Figure 1 or the external memory 106 in Figure 9. The means for accessing a data value corresponds to a data access instruction executed either by the processor core 6 in Figure 1, or the instruction decoder 118 and other elements in the microprocessor integrated circuit 104 shown in Figure 9. The corresponding structure in Figure 1 is described on page 8, lines 4-21. The corresponding structure in Figure 9 is described on page 13, lines 4-32.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The primary rejection, and the rejection to be reviewed on appeal, is the rejection of claims 1-4, 7, 12-17, and 20 under 35 U.S.C. §103 as being unpatentable over the description of the prior art, (referred as to "DPA"), in view of USP 6,681,321 to Dale et al. The other secondary rejections do not need to be addressed on appeal because the primary rejection is in error.[1]

VII. ARGUMENT

A. The Combination of Dale and the DPA Fails to Disclose or Suggest Every Feature in Claims 1, 12, and 14

The Examiner contends that the DPA discloses all features in the independent claim except for "generating a data place holder or insertion of late data into a data stream." The Examiner asserts that Dale remedies these deficiencies. Appellants disagree.

As explained, generating a trace data stream representing the step-by-step activity of a processor is troublesome when completing one instruction is not required before starting execution of a subsequent instruction. Recall the data "miss" examples provided above in Figure 3 where a data value is not present in cache memory and must be retrieved from the main memory, thereby incurring a variable delay of possibly several processing cycles. Although the processor continues to generating program instructions while awaiting data from a previous load miss, the problem is how to provide a

meaningful trace data stream that enables correlation of the data being recovered later from memory with the earlier executed instructions. The inventors solved this problem by (1) providing a *data place holder* in the trace data stream *in response to a data miss*, and (2) inserting an identifier for the data value at a later point in the trace data stream when a successful access to the missing data value has been made. The place holder and identifier can then be matched together.

Dale describes an apparatus for instruction execution tracing with out-of-order, speculative processors. As explained at column 5, beginning at line 9, instructions and data are loaded into their respective caches 406 and 404, and address and content information for those instructions and data are captured by a trace buffer 490. This buffered information is then sent to and stored in a trace storage 495, and that stored data is used to construct a "snap shot" of the caches 406 and 404.

Independent claim 1 recites that the "tracing circuit is *responsive to said data miss to generate a data place holder within said stream of trace data* at a position where data identifying said data value would have been placed if said data misses had not occurred." The Examiner identifies the "unique identifier information" in column 2, lines 16-19 in Dale as corresponding to the claimed data place holder. This correspondence cannot be reasonably maintained.

Contrary to being "responsive to a data miss," as recited in claim 1, Dale teaches in column 6, lines 42-45 that the unique identifier is created and sent to the trace buffer as instructions are "sequenced by the sequencing unit 450." As each instruction is

[1] Appellants reserve the right to argue against these rejections in subsequent proceedings should the Board come to a different conclusion.

sequenced by the sequencing unit 450, a unique identifier associated with each (as yet unexecuted) instruction is stored in the trace storage device 495 along with the instruction address. In other words, the unique identifier is created, not in response to a data miss, but instead simply as a result of instructions being sequenced for execution. Dale's identifiers are used to track execution of each instruction to create a sequential execution list later used to reconstruct the instruction stream. Thus, Dale actually *teaches away* from what is recited in claim 1 in that Dale's data place holder (read onto the "unique stream identifier" in Dale) is generated *prior to execution* of the processing instruction at the instruction sequencing stage and **not** in response to a data miss resulting from an instruction execution.

Indeed, the Examiner concedes that Dales does not expressly disclose the quoted claim feature in paragraph 4 of the final action. Nonetheless, the Examiner argues that the text at column 2, lines 7-10 of Dale teaches this claimed feature because Dale responds to a data miss (in this case a cache miss) by updating cache information stored in the trace storage device. This text from Dale reads: "when a cache load is necessary to obtain instructions that are not already stored in the instruction cache or data that is not already stored in the data cache, updated cache information is stored in the trace storage device." This passage just discloses that Dale generates *some* response to a data miss. But the passage does not disclose or suggest the *particular response* to a data miss recited in claim 1—generating a data place holder in response to a data miss.

Instead, Dale simply teaches that trace data corresponding to cache update information is stored in a trace storage device in response to a cache miss. But the Board should not be confused. The Examiner has already identified Dale's "unique identifier"

as the alleged counterpart of the claimed data place holder—not the cache update trace data. And Dale's *unique identifiers have already been generated* during instruction sequencing—*prior to any data miss occurring*.

The Examiner contends that Appellants earlier conceded in the previous response that Dale's unique identifiers are placed sequentially in the trace data stream at "a position where data identifying said data value would have been placed had said data miss not occurred." This is not the case. Page 10 of the previous response referred to column 5, lines 38-44 of Dale which teaches that address information from fetched instructions is stored to the trace storage device in sequential order and that this stored data may subsequently be used to reconstruct the instruction stream of an executed program. Although this passage of Dale discloses that address information from fetched instructions is sequentially stored, it does not disclose that a data place holder (which Examiner contends is the unique identifier in Dale) is stored sequentially in the data stream at a position where data identifying the missed data value would have been placed had the data miss not occurred.

Accordingly, even if one were to combine DPA and Dale, that combination would not teach a tracing circuit generating a data place holder within a trace stream in response to a data miss, where the data miss occurred in response to execution of a data access instruction. Independent claims 12 and 14 contain similar claim features neither disclosed nor suggested by the combination of DPA and Dale. On this ground alone, the final rejections should be reversed since they all depend on the improper rejection based on DPA and Dale.

B. The Claimed Data Place Holder Is Generated In Response to a Data Miss and Not For Every Processing Instruction

An important distinction between the independent claims and Dale is that the data place holder is generated *in response to a data miss* and inserted at a particular position in the trace data stream. In contrast, Dale generates a unique identifier (1) prior to instruction execution at an instruction sequencing stage (2) for each and every processing instruction. This is significant in the context of the inventive step of the present invention.

Dale stores instruction address information for **all** fetched instructions and data in sequential order in the trace storage device 495. See column 5, lines 35-38. Dale later reconstructs the entire sequence of fetched instructions from the sequentially stored information in the trace storage device 495. See, for example, column 5, lines 38-44:

> In this way, a real time sequential listing of all instructions fetched by the fetch unit 420 may be obtained. This sequential listing may be used to reconstruct the instruction stream of an executed program, i.e. the particular instruction sequence or code created by the programmer of the program.

Because only data misses generate data place holders, much less trace data needs to be correlated than in Dale's approach where information for every fetched instruction is stored in the trace storage device 495 and used to reconstruct each instruction of an executed program. In addition, the combination of the data place holder and the late data value enables subsequent analysis of the trace data stream *per se* to correctly identify which data values correspond to which access instructions and to obtain a proper understanding of the data processing system (see e.g., the description page 5, lines 3-8).

It is much easier to correlate data values being returned from memory with instructions being executed in data processing systems that support continued operation following data access misses. That easy correlation is made possible by generating a data place holder in a trace data stream in response to a data miss and inserting, at a later point in the trace data stream, a late data value identifying the data associated with the data miss.

Dale's approach is quite different. First, Dale generates the unique identifiers prior to instruction execution at an instruction sequencing stage. A unique identifier is generated for each and every sequenced instruction rather than only for executed data access instructions associated with a data miss. Second, the Dale approach requires maintaining separate data structures relating to cache state changes on the one hand (the data structure 620 in Figure 6A) and instruction stream event information on the other hand (the data structure 640 in Figure 6B). Third, in column 8, line 51 to column 9, line 9, Dale requires that both cache model information and instruction stream event information be used in order to reconstruct the instruction stream.

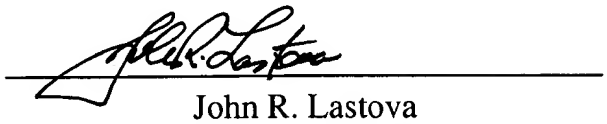
So Dale's approach involves much more burdensome correlation and analysis to reconstruct the instruction stream. This is because a unique identifier (which Examiner equates with the claimed data place holder) for each sequenced instruction must be tracked and correlated with cache status information. Furthermore, the Dale system requires two different trace data structures to be maintained in order to perform the instruction stream reconstruction, i.e., the cache status information and the instruction stream event information. These advantages achieved by the claimed invention, lacking in Dale's approach, are further evidence of non-obviousness.

VIII. CONCLUSION

Lacking features of the claims as explained above, the Board should reverse the outstanding rejections.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 

John R. Lastova
Reg. No. 33,149

JRL/sd
Appendix A - Claims on Appeal

IX. CLAIMS APPENDIX

1. Apparatus for processing data, said apparatus comprising:

(i) a processing circuit operable to process data values under control of processing instructions;

(ii) a memory operable to store data values to be processed, said processing circuit being responsive to a data access instruction to access a data value stored within said memory; and

(iii) a tracing circuit operable to generate a stream of trace data identifying processing instructions executed and data values accessed by said processing circuit; wherein

(iv) a data access instruction may result in a data miss such that a data value corresponding to said data access instruction is accessed upon a processing cycle subsequent to that upon which said access would occur without said data miss; and

(v) said tracing circuit is responsive to said data miss to generate a data place holder within said stream of trace data at a position where data identifying said data value would have been placed if said data miss had not occurred and then, when said access to said data value does occur, to insert at a later point in said stream of trace data a late data value identifying said data value.

2. Apparatus as claimed in claim 1, wherein said memory comprises a cache memory and a main memory, a data miss occurring when a data value being accessed is not stored within said cache memory.

3. Apparatus as claimed in claim 1, wherein said data place holder includes a tag value and said late data value includes a matching tag value.

4. Apparatus as claimed in claim 3, wherein when a plurality of data misses result in a plurality of data place holders being generated, late data values may be generated in a different order than their corresponding data misses.

5. Apparatus as claimed in claim 1, wherein said data place holder includes data identifying how many outstanding late data values are awaited at that time.

6. Apparatus as claimed in claim 1, wherein said stream of trace data includes periodic synchronizing data, said synchronizing data including data identifying how many outstanding late data values are awaited at that time.

7. Apparatus as claimed in claim 1, wherein said tracing circuit is operable to control tracing operation in response to a trigger condition associated with one or more of said data value and a memory address associated with said data value.

8. Apparatus as claimed in claim 7, wherein said tracing circuit is responsive to an exact match signal such that a trigger condition associated with a data value for which a data miss occurs is either:

(i) not triggered until said data value is accessed and found to meet said trigger condition; or

(ii) triggered upon said data miss upon an assumption that said data value when accessed will meet said trigger condition.

9. Apparatus as claimed in claim 8, wherein said exact match signal is user configurable.

10. Apparatus as claimed in claim 8, wherein said exact match signal is set under hardware control depending upon a use of said trigger condition.

11. Apparatus as claimed in claim 8, wherein said exact match signal has different values in different parts of said tracing circuit to simultaneously provide behaviors whereby said

tracing circuit is responsive to an exact match signal such that a trigger condition associated with a data value for which a data miss occurs is either:

(i) not triggered until said data value is accessed and found to meet said trigger condition; or

(ii) triggered upon said data miss upon an assumption that said data value when accessed will meet said trigger condition.

12. A method of processing data, said method comprising the steps of:

(i) processing data values under control of processing instructions;

(ii) storing data values to be processed, a data access instruction being operable to access a stored data value; and

(iii) generating a stream of trace data identifying processing instructions executed and data values accessed; wherein

(iv) a data access instruction may result in a data miss such that a data value corresponding to said data access instruction is accessed upon a processing cycle subsequent to that upon which said access would occur without said data miss; and

(v) in response to said data miss, generating a data place holder within said stream of trace data at a position where data identifying said data value would have been placed if said data miss had not occurred and then, when said access to said data value does occur, inserting at a later point in said stream of trace a late data value identifying said data value.

13. A computer program product carrying out a computer program for controlling a data processing apparatus to analyze a stream of trace data generated in accordance with the method of claim 12.

14. Apparatus for processing data, comprising:

means for processing data values under control of processing instructions;

means for storing data values to be processed, said processing means being responsive to a data access instruction to access a data value stored within said memory means;

means for generating a stream of trace data identifying processing instructions executed and data values accessed by said processing circuit; and

means for accessing a data value corresponding to a data access instruction resulting in a data miss at a processing cycle subsequent to that at which said access would occur without said data miss;

wherein the means for generating is responsive to said data miss to generate a data place holder within said stream of trace data at a position where data identifying said data value would have been placed if said data miss had not occurred and the, when said access to said data value does occur, to insert at a later point in said stream of trace data a late data value identifying said data value.

15. Apparatus as claimed in claim 14, wherein said means for storing comprises a cache memory and a main memory, a data miss occurring when a data value being accessed is not stored within said cache memory.

16. Apparatus as claimed in claim 14, wherein said data place holder includes a tag value and said late data value includes a matching tag value.

17. Apparatus as claimed in claim 14, wherein when a plurality of data misses result in a plurality of data place holders being generated, late data values may be generated in a different order than their corresponding data misses.

18. Apparatus as claimed in claim 14, wherein said data place holder includes data identifying how many outstanding late data values are awaited at that time.

19. Apparatus as claimed in claim 14, wherein said stream of trace data includes periodic synchronizing data, said synchronizing data including data identifying how may outstanding late data values are awaited at that time.

20. Apparatus as claimed in claim 14, wherein said means for generating is a tracing circuit operable to control a tracing operation in response to a trigger condition associated with one or more of said data value and a memory address associated with said data value.

21. Apparatus as claimed in claim 20, wherein said means for generating is responsive to an exact match signal such that a trigger condition associated with a data value for which a data miss occurs is either:

(i) not triggered until said data value is accessed and found to meet said trigger condition; or

(ii) triggered upon said data miss upon an assumption that said data value when accessed will meet said trigger condition.

22. Apparatus as claimed in claim 21, wherein said exact match signal is user configurable.

23. Apparatus as claimed in claim 21, wherein said exact match signal is set under hardware control depending upon a use of said trigger condition.

24. Apparatus as claimed in claim 21, wherein said exact match signal has different values in different parts of said means for generating to simultaneously provide behaviors whereby said means for generating is responsive to an exact match signal such that a trigger condition associated with a data value for which a data miss occurs is either:

(i) not triggered until said data value is accessed and found to meet said trigger condition; or

(ii) triggered upon said data miss upon an assumption that said data value when accessed will meet said trigger condition.

X. EVIDENCE APPENDIX

There is no evidence appendix.

XI. RELATED PROCEEDINGS APPENDIX

There is no related proceedings appendix.